# Test Plan **Karl Erb, Christopher Kheir, Anthony Chambers, Zachary Baas**

## 1.1. Testing Team Organization and Resource Assignments

**Table 1.1 – Team Organization**

|  |  |
| --- | --- |
| **Team Member** | **Roles and Responsibilities** |
| Christopher Kheir | Hardware & Operating System Configuration (0)  Responsibilities:   * Hardware configuration * Test internet connection * Install Linux Ubuntu configuration   Estimated Effort:   * Significant effort is required for testing hardware configuration. * Minimal effort is required for testing internet connection. * Minimal effort is required for testing the installation of Ubuntu.   This role will be primarily responsible for the test cases beginning with “0” in Section 1.3. |
| Karl Erb | Web Development (1)  Responsibilities:   * Leading the team for the project * Installation and configuration of the Apache web server * Implementing and testing the web client design   Estimated Effort:   * Moderate effort required for testing the web client design * Minimal effort required for testing the Apache web server   This role will be primarily responsible for the test cases beginning with “1” in Section 1.3. |
| Zach Baas | Database Management (2)  Responsibilities:   * Installing MySQL * Installing PHP * Configuring MySQL & PHP settings * Database setup * Load sample data * Ensure communication between the database and web site via PHP * Testing the database with the feature requirements   Effort Required   * Effort required will be minimal for testing, as most of the testing involves opening test pages and confirming that the results are done. * More effort or backend knowledge would be required developer-side to test the queries themselves in the phpMyAdmin backend.   This role will be primarily responsible for the test cases beginning with “2” in Section 1.3. |
| Anthony Chambers | Security (3)  Responsibilities:   * Penetration Testing * Being Compliant with FERPA * Installation of Firewalls * Implementation of HTTPS * Enabling and configuring CAPTCHA   Effort Required:   * Effort required for testing will be minimal with the exception for Penetration testing. Most of the testing will require opening and accessing pages. * More effort will be required for creating and tracking users within the system and finding exploits through penetration testing.   This role will be primarily responsible for the test cases beginning with “3” in Section 1.3. |

***1.2. Testing Schedule***   
  
This section is not applicable to this project, as no working code is being developed for the MVC SIS prototype assignment.  
***1.3. Test Cases***   
**Table 1.3 – Test Cases Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Requirement Number** | **Test Goal** | **Test Method/Technique** |
| 000 | 1 | To test if the hardware is capable of powering on and running software | White Box Testing – Compatibility Testing |
| 001 | 1 | To test that the system is connected to the internet | Black Box Testing – Compatibility Testing |
| 002 | 2.1 | To test if Linux Ubuntu is installed correctly | Black Box Testing – Compatibility Testing |
| 100 | 3.1 | To test if the home page is displayed properly | Black Box Testing –  Usability Testing,  GUI Testing |
| 101 | 3.1 | To test if the login button takes the user to the login screen | Black Box Testing –  Functional Testing, |
| 102 | 3.2 | To test if the login screen is displayed properly | Black Box Testing –  Usability Testing,  GUI Testing |
| 103 | 3.2 | To test if the username is validated correctly | Black Box Testing –  Functional Testing |
| 104 | 3.2 | To test if the password is validated correctly | Black Box Testing –  Functional Testing |
| 105 | 3.2 | To test if the user is able to successfully log into the system | Black Box Testing –  Functional Testing |
| 106 | 3.3 | To test if the welcome screen is displayed properly | Black Box Testing –  Usability Testing,  GUI Testing |
| 107 | 3.4 | To test if the student welcome screen is displayed properly and only shows the button for students to view or edit their record | Black Box Testing –  Usability Testing,  GUI Testing |
| 108 | 3.4 | To test if the edit student button takes the student user to the edit student screen | Black Box Testing –  Functional Testing |
| 109 | 3.5 | To test if the faculty welcome screen is displayed properly shows only the buttons for faculty users to search or edit student records | Black Box Testing – Usability Testing,  GUI Testing |
| 110 | 3.5 | To test if the search student button takes the faculty user to the search student record screen | Black Box Testing –  Functional Testing |
| 111 | 3.5 | To test if the edit student button takes the faculty user to the edit student record screen | Black Box Testing –  Functional Testing |
| 112 | 3.6 | To test if the registrar staff welcome screen is displayed properly and shows the buttons for registrar staff users to navigate the system | Black Box Testing –  Usability Testing,  GUI Testing |
| 113 | 3.6 | To test if the search student button takes the registrar staff user to the search student record screen | Black Box Testing –  Functional Testing |
| 114 | 3.6 | To test if the edit student button takes the registrar staff user to the edit student record screen | Black Box Testing –  Functional Testing |
| 115 | 3.6 | To test if the add student button takes the registrar staff user to the add student record screen | Black Box Testing –  Functional Testing |
| 116 | 3.6 | To test if the delete student button takes the registrar staff user to the delete student record screen | Black Box Testing –  Functional Testing |
| 117 | 3.6 | To test if the assign user permissions button takes the registrar staff user to the assign user permissions screen | Black Box Testing –  Functional Testing |
| 118 | 3.6 | To test if the view transaction logs button takes the registrar staff user to the view transaction logs screen | Black Box Testing –  Functional Testing |
| 119 | 3.10 | To test if the view student info option for student users only displays the user’s own record | Black Box Testing –  Usability Testing,  GUI Testing |
| 120 | 3.9 | To test if the view student info option is displayed properly for faculty users | Black Box Testing –  Usability Testing,  GUI Testing |
| 121 | 3.9 | To test if the view student info option is displayed properly for registrar staff users | Black Box Testing –  Usability Testing,  GUI Testing |
| 122 | 3.7 | To test if the search student records page is displayed correctly for faculty users and allows them to search for student records | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 123 | 3.8 | To test if the search student records page is displayed correctly for registrar users and allows them to search for student records | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 124 | 3.11 | To test if the edit student info option is displayed properly for student users and only allows them to edit their own record | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 125 | 3.12 | To test if the edit student info option is displayed properly for faculty users and allows them to edit student records | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 126 | 3.13 | To test if the edit student info option is displayed properly for registrar staff users and allows them to edit student records | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 127 | 3.14 | To test if the add student record option for registrar staff users is displayed properly and allows them to add new student records | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 128 | 3.15 | To test if the delete student record option for registrar staff users is displayed properly and allows them to delete student records | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 129 | 3.16 | To test if the assign roles and permissions option for registrar staff users is displayed properly and allows them to assign roles and permissions | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 130 | 3.17 | To test if the transaction logging option for registrar staff users is displayed properly and allows them to search for transaction logs | Black Box Testing –  Functional Testing,  Usability Testing,  GUI Testing |
| 131 | 4.1 | To test if the web client can successfully connect to the Apache server | White Box Testing –  Integration Testing  Non-functional Testing – Performance Testing |
| 200 | 5.1 | Have a working MySQL connection established with proper credentials and port access. | Black Box Testing – Functional Testing |
| 201 | 5.2 | Have databases created and confirm access to the database. | Black Box Testing – Functional Testing |
| 202 | 5.3 | Have tables created within the databases that match the ERD diagrams. | Black Box Testing – Functional Testing |
| 203 | 5.4 | Have sample student data added that is sufficient for end user testing. | Black Box Testing – Functional Testing |
| 204 | 7.1 | Be able to connect to and query the MySQL database using PHP from the web interface. | Black Box Testing –  Functional Testing |
| 205 | 8.1 | Successfully add new students | Black Box Testing –  Functional Testing |
| 206 | 8.2 | Successfully query the table to display specific students. | Black Box Testing –  Functional Testing |
| 207 | 8.3 | Successfully delete students. | Black Box Testing –  Functional Testing |
| 208 | 8.4 | Successfully change student data. | Black Box Testing –  Functional Testing |
| 209 | 8.5 | Successfully display student data. | Black Box Testing –  Functional Testing |
| 210 | 8.6 | Successfully throw an error if there is an attempted duplication of the SID | Black Box Testing –  Functional Testing |
| 211 | 8.7 | Successfully requires user to commit action | Black Box Testing –  Functional Testing |
| 300 | 6.1 | Installing and configuring a functioning firewall for the database | Penetration Testing  Security Testing |
| 301 | 6.2 | Conduct penetration testing on the database to test the security. | Security Testing -  Penetration Testing  Vulnerability Testing |
| 302 | 6.3 | Successfully implementing and purchasing security certificates for HTTPS | Security Testing |
| 303 | 6.4 | Enable logging and using it to track access to the SIS | Security Testing |
| 304 | 6.5 | Successfully installing CAPTCHA plugin to prevent automatic logins | Security Testing |
| 305 | 6.6 | Being in the compliance with FERPA | Security Testing-  User Acceptance Testing |

## 1.1. Testing Team Organization and Resource Assignments

State the members of the test team, their roles and responsibilities, and the estimated effort required of each team member, and for the team as a whole.

This needs to have a ***good level of detail*** to show the testing process has been thought through and the effort properly planned and allocated to team members.

|  |  |
| --- | --- |
| **Team Member** | **Roles and Responsibilities** |
|  |  |

***1.2. Testing Schedule***  
***If no code has been written*** – this section can be labelled N/A.

***If a Working Application was developed*** --State the testing and debugging schedule, including planned releases from the programmers to the testers for updated versions of the system, delivery deadlines, and milestones.

This needs to include ***team scheduling*** as well as the overall project deadlines.

Display the Schedule in a ***Gantt chart and embed in the document here. Also submit a separate file used to create the Gantt chart with your STS submission***.

***1.3. Test Cases***  
List all test cases used in the testing effort.

This would include ***all the requirements*** found in the Updated Requirements Traceability Matrix.

Typically, **each requirement** should have its own test case, however multiple requirements can be put in the same test case but provisions must be made to indicate the success or failure of each individual requirement.

Test cases should be organized into black box and white box tests, and arranged to make clear how equivalence partitioning, boundary value analysis, coverage goals, and so forth, were used to generate the test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Requirement Number** | **Test Goal** | **Test Method/Technique** |
| Any numbering system, i.e. one-up | Maps the requirements in the Traceability Matrix to specific test case(s). Also  create design-driven test cases. | Short description of the test goal(s) and objective(s). | [Indicate the test method & technique](https://www.dropbox.com/s/rhj9bbyj5qmigx3/2c-Types-of-software-testing-complete-list.pdf?dl=0) [**for example -** Black box - Equivalence Partitioning, etc.] |